

# 4D\_Pack

---

*Reference Guide*  
*Windows® / Mac OS®*



---

## **4D\_Pack - Reference Guide**

### **Version 11 for Windows® and Mac OS®**

Copyright © 4D SAS/4D, Inc. 1985-2009  
All rights reserved.

---

The Software described in this manual is governed by the grant of license in the 4D Product Line License Agreement provided with the Software in this package. The Software, this manual, and all documentation included with the Software are copyrighted and may not be reproduced in whole or in part except for in accordance with the 4D Product Line License Agreement.

4th Dimension, 4D, the 4D logo, and 4D Server are registered trademarks of 4D, Inc.

Microsoft and Windows are registered trademarks of Microsoft Corporation.

Apple, Macintosh, Mac OS and QuickTime are trademarks or registered trademarks of Apple Computer, Inc.

Mac2Win Software Copyright © 1990-2009, is a product of Altura Software, Inc. This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

4D includes cryptographic software written by Eric Young ([ey@cryptsoft.com](mailto:ey@cryptsoft.com))  
4D includes software written by Tim Hudson ([tjh@cryptsoft.com](mailto:tjh@cryptsoft.com)).

ACROBAT © Copyright 1987-2009, Secret Commercial Adobe Systems Inc. All rights reserved. ACROBAT is a registered trademark of Adobe Systems Inc.

4D\_Pack is based in part on the work of the Independent JPEG Group and the Quest Center at Cold Spring Harbor Labs.

All other referenced trade names are trademarks or registered trademarks of their respective holders.

# Contents

## 1. Preface..... 5

4D\_Pack, Introduction.....7

## 2. 4D\_Pack : User Interface..... 9

AP Rect dragger..... 11

AP FULL SCREEN..... 12

AP NORMAL SCREEN..... 13

## 3. 4D\_Pack : Windows Help Files.. 15

AP HELP ON KEY..... 17

AP HELP INDEX..... 18

AP HELP ON HELP..... 19

AP CLOSE HELP..... 20

## 4. 4D\_Pack : Utilities..... 21

AP ShellExecute.....23

AP Get table info.....25

AP Get field infos.....26

AP Get templates.....28

AP Create method.....29

AP Does method exist..... 31

AP Modify method..... 32

AP Get tips state.....33

AP SET TIPS STATE.....34

AP Get file MD5 digest..... 35

AP Timestamp to GMT..... 36

AP Get picture type..... 37

AP AVAILABLE MEMORY..... 38

AP GET PARAM.....39

AP SET PARAM..... 40

## **5. 4D\_Pack : External Clock..... 45**

_AP External clock.....	47
AP SET CLOCK.....	48

## **6. 4D\_Pack : Printing..... 49**

AP Print settings to BLOB.....	51
AP BLOB to print settings.....	52

## **7. 4D\_Pack : ANSI Streams..... 55**

AP fopen.....	57
AP FPRINT.....	59
AP FCLOSE.....	60
AP fread.....	61

## **8. 4D\_Pack : Picture Files..... 63**

AP Save BMP 8 bits.....	65
-------------------------	----

## **Command Index..... 67**

# 1

---

# Preface



The 4D\_Pack plug-in provides 4D developers with useful additional routines.

The 4D\_Pack command names have the “AP” prefix in order to avoid any confusion with 4D commands. They also use the uppercase/lowercase convention depending on whether they are commands or functions.

### Installation

You install the 4D\_Pack plug-in just as any 4D plug-in: you must put it in the **Plugins** folder located either next to the executable application or next to the structure file of the database. For more information, please refer to the *Design Reference* manual.

### Note to users of previous versions of 4D\_Pack

---

For reasons related to the evolution of 4D products, several commands found in previous versions have been removed in 4D\_Pack v11. Here is the list of deleted commands as well as the recommended replacement solutions:

AP PICT DRAGGER -> Use the integrated drag and drop functions of 4D (MouseDown, MouseX and MouseY System Variables in the case of “click” type events)  
AP PICT UPDATER and %AP Pict displayer -> Use a picture field  
AP Read Picture BLOB -> Use BLOB TO PICTURE  
AP Read Picture File -> Use READ PICTURE FILE  
AP Save GIF -> Use CONVERT PICTURE + WRITE PICTURE FILE  
AP SET PICT MODE -> Use SET FORMAT  
AP Select document -> Use Select document  
AP Set palette -> Use Select RGB Color  
AP Sublaunch -> Use LAUNCH EXTERNAL PROCESS  
AP PrintDefault -> Use AP BLOB to print settings + AP Print settings to BLOB  
AP Text to PrintRec -> Use AP BLOB to print settings + AP Print settings to BLOB  
AP PrintRec to text -> Use AP BLOB to print settings + AP Print settings to BLOB  
AP PrValidate -> Use AP BLOB to print settings + AP Print settings to BLOB  
AP Toolbar installed -> Use Tool bar height (returns 0 if toolbar is hidden)  
AP SET WEB FILTERS -> Use SET DATABASE PARAMETER and Get database parameter  
AP ShellExecute (Mac OS) → Use LAUNCH EXTERNAL PROCESS  
AP Add table and fields -> Use 4D SQL engine  
AP Create relation -> *No alternative solution currently*





# 2

---

## 4D\_Pack : User Interface



---

AP Rect dragger (left; top; right; bottom; x; y) → Integer

Parameter	Type		Description
left	Longint	→	Left coordinate of the visible rectangle
top	Longint	→	Top coordinate of the visible rectangle
right	Longint	→	Right coordinate of the visible rectangle
bottom	Longint	→	Bottom coordinate of the visible rectangle
x	Longint	←	Horizontal location of the mouse
y	Longint	←	Vertical location of the mouse
Function result	Integer	←	Process ID of the process in which the item was dragged

### Description

AP Rect dragger allows you to implement a “Drag and Drop” interface within 4D. It displays a graphical representation of an item being dragged from one place to another and returns the coordinates of the mouse release and the process ID of the process in which the item was dragged.

### See also

CALL PROCESS.

**AP FULL SCREEN**

<b>Parameter</b>	<b>Type</b>	<b>Description</b>
		This command does not require any parameters

**Description**

The AP FULL SCREEN command enables you to hide the menu bar and the title bar of the 4D application window and to set this window to full-screen size.

This command only works on Windows.

**See Also**

AP NORMAL SCREEN.

**AP NORMAL SCREEN**

<b>Parameter</b>	<b>Type</b>	<b>Description</b>
------------------	-------------	--------------------

This command does not require any parameters

**Description**

The AP NORMAL SCREEN command enables you to resize a menu bar and a title bar that were previously hidden by the AP FULL SCREEN command.

This command only works in Windows.

**See Also**

AP FULL SCREEN.



# 3

---

## 4D\_Pack : Windows Help Files





**Compatibility Note:** This command is not compatible with Windows Vista. Its use is no longer recommended.

AP HELP ON KEY (help; keyword)

<b>Parameter</b>	<b>Type</b>	<b>Description</b>
help	String	→ Name of the Help file
keyword	String	→ Keyword used to search for help

### **Description**

The AP HELP ON KEY command opens a Help file whose name is passed as a parameter. This command uses the standard Windows Help utility, which displays a page corresponding to the keyword found (on Windows only).

**Compatibility Note:** This command is not compatible with Windows Vista. Its use is no longer recommended.

AP HELP INDEX (help)

<b>Parameter</b>	<b>Type</b>	<b>Description</b>
help	String	→ Name of the Help file

### **Description**

The AP HELP INDEX command displays the list of keywords available in the Help file whose name was passed as a parameter (on Windows only).

**Compatibility Note:** This command is not compatible with Windows Vista. Its use is no longer recommended.

AP HELP ON HELP

<b>Parameter</b>	<b>Type</b>	<b>Description</b>
------------------	-------------	--------------------

This command does not require any parameters

**Description**

The AP HELP ON HELP command displays the on-line help for the program's own Help file (on Windows only).

**Compatibility Note:** This command is not compatible with Windows Vista. Its use is no longer recommended.

AP CLOSE HELP

Parameter	Type	Description
-----------	------	-------------

This command does not require any parameters

### Description

The AP CLOSE HELP command closes the Windows Help application (on Windows only).

# 4

---

## 4D\_Pack : Utilities



**Compatibility Note:** This command is now obsolete; it is kept only for compatibility reasons and will be removed in future versions of the plug-in. It is strongly recommended to use the LAUNCH EXTERNAL PROCESS command of 4D. Note that starting with version 11.0 of 4D, AP ShellExecute only works under Windows.

AP ShellExecute (fileName{; mode{; param}}) → Longint

Parameter	Type	Description
fileName	String	→ Name of file or document to be launched
mode	Longint	→ Status of window at time of launch
param	Text	→ Application parameters
Function result	Longint	← 0, if the command execution succeeded; otherwise, a system error

**Note:** This command only functions under Windows. Under Mac OS, it has no effect.

### Description

The AP ShellExecute command enables you to launch an application or open a document from a 4D database.

In the fileName parameter, pass the name or complete (absolute) access path of the file to be launched. When the application or document to be opened is located next to the database structure, it is possible to pass only the file name or a relative access path. Simply indicate a document name and Windows handles the execution of the associated application.

The mode parameter enables you to specify the display mode of the window in which the application will be launched:

- if you pass 0, the window will be normal;
- if you pass 1, the window will be full-screen;
- if you pass 2, the window will be reduced and will appear in the Windows task bar.

The param parameter enables you to pass any additional type of parameter that may be necessary for the launched application. For instance, if you use this command to launch a Web browser, you can pass a specific URL in this parameter.

## Examples

1. This example opens a Word® document located at the root of the main disk:

```
Under Windows  
$err:=AP ShellExecute ("C:\Test.doc")
```

2. This example launches an internet browser, in full-screen mode by default, on the 4D site:

```
$err:=AP ShellExecute ("www.4d.fr";"1")
```

3. This example opens the 4D site in full-screen on Netscape® whatever the default internet browser:

```
$err:=AP ShellExecute ("C:\Program Files\Netscape\NETSCAPE.EXE";"1";"www.4d.fr")
```

4. This example opens the Windows Explorer in a normal window and displays the contents of the Music folder:

```
$err:=AP ShellExecute ("C:\WINDOWS\explorer.exe";0;"C:\Music")
```

## See Also

LAUNCH EXTERNAL PROCESS.



AP Get table info (table; invisible; destruct) → Integer

Parameter	Type		Description
table	Integer	→	Table number
invisible	Integer	←	1, if the table is invisible 0, if the table is visible
destruct	Integer	←	1, for physical deletion of the records 0, for logical deletion of the records
Function result	Integer	←	Error code

**Description**

AP Get table info returns information on 4D tables.

Given the table number, the invisible parameter is set to 1 if the table is declared as invisible in Design mode, otherwise it is set 0.

The destruct variable is set to 1 for physical deletion of the records, or 0 for logical deletion. The physical deletion of the records is slower than the logical one but it prevents the record from reappearing in case of a "recover by tags" operation.

AP Get field infos (table; field; relTable; relFld; props; choiceList) → Integer

Parameter	Type		Description
table	Integer	→	Table number
field	Integer	→	Field number
relTable	Integer	←	Related table
relFld	Integer	←	Related field
props	Longint	←	Field properties
choiceList	String	←	Choice list
Function result	Integer	←	Error code

**Description**

AP Get field infos returns specific information about a field.

Given the table number and field number, the command returns the linked table and the linked field number, respectively.

If there is no relation with the given field, relTable and relFld are set to 0.

The 16 bits of props are meaningful. Each bit can be read via bits operators of 4D language.

Bit number:	Is set to 1 if:	Ignored if field type is:
15	Indexed	Text, Picture, Subtable or Blob
14	Indexed unique	Text, Picture or Subtable or Blob or if bit 15 is set to 0
13	Mandatory	Subtable
12	Choice list	Picture, Subtable or Blob
11	Modifiable	Subtable
10	Enterable	Subtable
8	Invisible	
6	N to 1 relation	Text, Picture, Subtable or Blob
5	1 to N relation	Text, Picture, Subtable or Blob

2 and 1	Automatic 1 to N relation whose deletion control is: 00 No deletion control 01 No assignment of related records 10 Deletion of related records 11 Deletion of related records not allowed	
0	Automatic N to 1 relation with automatic assignment	Text, Picture, Subtable, Blob, or if bit 5 not set

**Note:** Bits 3, 4, 7 and 9 are reserved for future use.

If bit 12 is set, the variable choiceList contains the name of the list.

AP Get templates (templateNameArray)

Parameter	Type	Description
templateNameArray	String Array ←	Array of template names

### Description

The AP Get templates command populates the array templateNameArray with the list of the form templates available in the current 4D application.

The form templates can be selected in the Form Wizard dialog box. Available built-in templates are for example Nostalgia, OSX, OSX (no labels), Philadelphia, etc. Customized forms built in the Advanced pages of the Form Wizard can be added as templates.

This command is useful when you use the AP Add table and fields command, because you can determine the template to use for the default forms.

**Note:** This command does not work when it is executed from an application that has been compiled and merged 4D Desktop.

If the command has been executed properly, it returns 0. Otherwise, it returns an error code.

### Example

This example creates a window containing all the available templates:

```
$ref:=Open window(<>WLeft;<>WTop;450;400;8;"Template List")
ARRAY STRING(255;ArrTemplates;0)
$error:=AP Get templates (ArrTemplates)
DIALOG([Table 1];"ArrTemplates")
CLOSE WINDOW($ref)
```

AP Create method (methodName; propertiesArray; methodCode{; folderName}) → Longint

Parameter	Type	Description
methodName	String (31)	→ Name of the method (31 chars)
propertiesArray	Longint Array	→ Array of properties (7 elements)
methodCode	BLOB	→ BLOB containing the method text
folderName	String	→ Explorer folder in which the method will be created
Function result	Longint	← Error code (0=no error)

### Description

The AP Create method command lets you add a project method in a 4D database structure (interpreted databases only).

Pass the name of the method in `methodName`. This name can be up to 31 characters long. It can contain any combination of letters, numbers, spaces and underlines with respect to the standard 4D objects naming rules. If the name contains more than 31 characters, it will be truncated to 31.

The `propertiesArray` parameter allows defining the method properties. It must have been declared previously as a longint array and must contain 4 items:

- pass 1 in `propertiesArray{1}` if the method must be visible and 0 if not.
- pass 1 in `propertiesArray{2}` if the method must be available for 4DACTION, 4DMETHOD and 4DSCRIPT, and 0 if not.
- pass 1 in `propertiesArray{3}` if the method must be offered as a Web Service and 0 if not.
- in case of `propertiesArray{3}=1`, pass 1 in `propertiesArray{4}` if the method must be published in WSDL and 0 for the other cases.
- pass 1 in `propertiesArray{5}` if the method must be shared between components and the host database, and 0 in the opposite case.
- pass 1 in `propertiesArray{6}` if the method must be available via SQL, and 0 in the opposite case.
- pass 1 in `propertiesArray{7}` if the method must have the "Execute on Server" attribute, and 0 in the opposite case.

Pass a BLOB containing the text of the method in `methodCode`. If you use the TEXT TO BLOB command to fill the BLOB, pass Mac C string (or 0) in the third parameter of this command.

In `methodCode`, you can pass the text of a method that has been stored in a file via the **Export Method...** menu command (in this case, the text includes the commands, constants, etc. as references (tokens) and not words).

In `folderName`, pass the name of the Explorer folder in which the method will be created. These folders are managed on the Home page of the Explorer window and allow you to organize the objects in a customized manner. This parameter is optional; if it is omitted, the method will be created in the “Default Project Methods” folder.

If the operation is completed successfully, the function returns 0. Otherwise, it returns an error code:

- 1 = A bad parameter type has been passed.
- 2 = The database is running in compiled mode.
- 3 = The method name is an empty string.
- 4013 = Invalid method name.
- 4014 = Could not create method.

**Note:** This command does not work when it is executed from a compiled application that has been merged with 4D Desktop.

### See Also

AP Does method exist, AP Modify method.

AP Does method exist (methodName) → Integer

Parameter	Type	Description
methodName	String	→ Name of method to test
Function result	Integer	← 0 = Method does not exist, 1 = Method already exists

### Description

The AP Does method exist command can be used to find out whether a project method named methodName already exists in the current database. This command does not take into account any project methods of the components installed in the database.

When using the AP Create method command, the purpose of this command is to avoid getting an error message when a method having the same name already exists.

### See Also

AP Create method.

AP Modify method (methodName; methodCode) → Longint

Parameter	Type	Description
methodName	String (31) →	Name of method to modify
methodCode	BLOB →	BLOB containing the method text
Function result	Longint ←	Error code (0=no error)

### Description

The AP Modify method command lets you modify the source code of a project method in the structure of a 4D database (interpreted databases only).

Pass the name of an existing project method in the `methodName` parameter (the names of 4D methods may contain up to 31 characters).

In `methodCode`, pass a BLOB containing the new text of the method. If you use the TEXT TO BLOB command to fill the BLOB, specify the Mac C string format (or 0) as the 3rd parameter of this command.

If the method is modified correctly, the command returns 0. Otherwise, it returns one of the following error codes:

- 1 = A bad parameter type has been passed.
- 2 = The database is running in compiled mode.
- 3 = The method name is an empty string.
- 4013 = Invalid method name.
- 4015 = The method source code could not be modified.

**Note:** This command does not work if it is executed from a compiled application.

### See Also

AP Create method, AP Does method exist.



AP Get tips state → Longint

Parameter	Type	Description
-----------	------	-------------

This command does not require any parameters

Function result	Longint	← Tip display status
-----------------	---------	----------------------

### Description

AP Get tips state returns 1 if tips are displayed; otherwise it returns 0.

### See Also

AP SET TIPS STATE.

AP SET TIPS STATE (status)

Parameter	Type	Description
status	Integer	→ Tips status to be set 0 = Do not display tips 1 = Display tips

### Description

The AP SET TIPS STATE command allows to display or not display tips.

### See Also

AP Get tips state.

AP Get file MD5 digest (filePath; digest{; fork}) → Longint

Parameter	Type		Description
filePath	Text	→	Complete pathname of file
digest	Text	←	MD5 Digest of file
fork	Longint	→	0=Data fork, 1=Resource fork
Function result	Longint	←	Error code

**Description**

The AP Get file MD5 digest command returns the MD5 digest key of a specific document. The MD5 (Message Digest 5) algorithm is a hashing function used for encrypting data.

Pass the name of the document containing the key in the filePath parameter. The MD5 digest key is returned in the digest parameter.

The fork parameter can be used to set, under Mac OS only, the part of the file to be selected:

- 0 for the data fork
- 1 for the resource fork.

**Example**

1. Typical use of command:

```

C_TEXT($thedoc)
C_TEXT(<>digest)
C_LONGINT($resfork)
$resfork:=0
$thedoc:=Select document
$error:=AP Get file MD5 digest($thedoc;<>digest;$resfork)
    
```

AP Timestamp to GMT (localDate; localTime; gmtDate; gmtTime) → String

Parameter	Type	Description
localDate	Date	→ Local date to use for the conversion
localTime	Time	→ Local time to use for the conversion
gmtDate	Date	← GMT date returned by the conversion
gmtTime	Time	← GMT time returned by the conversion
Function result	String	← GMT time in compliance with RFC standard

**Description**

AP Timestamp to GMT converts local date and time to GMT date and time based on the machine location (set with the Date/Time control panel). It returns a string in the following format:

"Weekday, DD MM YYYY HH:MM:SS GMT"

This format is in compliance with RFC 822 standard.

**Example**

Given the date of Thursday 11/13/97 and the time of 2:35PM (Paris time zone), the function returns the string "Thu, 13 Nov 1997 13:35:00 GMT". \$gmtd is set to 11/13/97 and \$gmtt to 13:35:00.

`$s:=AP Timestamp to GMT(Current date;Current time;$gmtd;$gmtt)`

**Compatibility Note:** This command is now obsolete; it is kept only for compatibility reasons and will be removed in future versions of the plug-in. It is strongly recommended to use the 4D commands of the "Pictures" theme.

AP Get picture type (storedPicture) → Longint

Parameter	Type	Description
storedPicture	Blob	→ Picture stored as a BLOB
Function result	Longint	← Type of the picture

**Description**

AP Get picture type returns the type of the image stored in the BLOB storedPicture:

- 1 Unknown type
- 0 'PICT' picture
- 1 QuickTime compressed 'PICT' picture
- 2 JPEG picture
- 3 WMF picture
- 4 EMF picture
- 5 BMP picture
- 6 GIF picture

**Example**

This example stores a picture in a BLOB and then stores its type in \$type.

```
DOCUMENT TO BLOB($thefile;$theblob)
$type:=AP Get picture type($theblob)
```

**Compatibility Note:** This command is now obsolete; it is kept only for compatibility reasons and will be removed in future versions of the plug-in.

AP AVAILABLE MEMORY (totalMemory; physicalMemory; freeMemory; freeStack)

Parameter	Type	Description
totalMemory	Longint	← Total memory
physicalMemory	Longint	← Physical memory
freeMemory	Longint	← Free memory
freeStack	Longint	← Free stack

### Description

The AP AVAILABLE MEMORY command returns information in Kilobytes on the installed and free memory on the machine where it is executed.

The command returns the machine installed memory (total and physical), the free memory available for 4D and the free stack (rounded to the closest Kb) for the current process.

**Compatibility Note:** This command is now obsolete; it is kept only for compatibility reasons and will be removed in future versions of the plug-in. It is strongly recommended to use the Get database parameter command of 4D.

AP GET PARAM (option; value)

Parameter	Type	Description
option	Integer	→ Number of the option to read
value	Integer	← Value for the option

### Description

AP GET PARAM allows you to read the internal 4D options usually set with 4D Customizer Plus. Using this command, you can restore a value set by 4D Customizer Plus when you want to temporarily change it using the command AP SET PARAM.

For more information about the value parameter, please refer to the AP SET PARAM command description.

### Example

The following code allows you to retrieve the value of the parameter 2:

```
AP GET PARAM(2;$param)
```

### See Also

AP SET PARAM.

**Compatibility Note:** This command is now obsolete; it is kept only for compatibility reasons and will be removed in future versions of the plug-in. It is strongly recommended to use the SET DATABASE PARAMETER command of 4D.

AP SET PARAM (option; value)

Parameter	Type	Description
option	Integer	→ Number of the option to set
value	Integer	→ New value for the option

### Description

The AP SET PARAM command allows you to set the internal 4D options usually set with 4D Customizer Plus. These options are stored in the CUST ID=0 resource which consists in a table of integer values numbered from 0 to n.

The integer values are as follows:

- 0 - If the value is 0, it is not possible to open a desk accessory while a 4D printing progress window is displayed. If the value is 1, it is possible to open a desk accessory when such a window is displayed. The default value is 0.
- 1 - If the value is 0, it is not possible to open a desk accessory while a sequential operation (search, sort...) progress window is displayed. If the value is 1, opening a desk accessory is possible while such a window is displayed. In the Windows version, the value -3 signifies that 4D does not use an offscreen bitmap to perform graphical updates to the screen. All drawing operations are directly performed on the screen. If the value is -4, at start-up 4D allocates a permanent offscreen buffer corresponding to the size and depth of your screen; it then uses this buffer to perform offscreen bitmap-based graphical updates. The default value is 0. Values other than those described specify that for each screen update 4D tries to allocate a corresponding buffer (provided there is enough memory to do so). If this is successful, 4D performs an offscreen bitmap-based update and then disposes of the buffer. Activating the offscreen bitmap-based updates allows you to eliminate screen blinking when many objects have to be redrawn.



A permanent offscreen buffer may take some space (i.e., 300K for a 640 x 480, 256 color screen), so for low memory configurations you may want to set this option to a value other than -3 or -4. Setting this option to -4 when memory is not an issue allows you to accelerate drawing operations. Deactivating offscreen bitmap-based updates (setting the option to -3) to take the screen updating scheme into account may solve incompatibility issues with 4D plug-ins that have not been updated under Windows.

- 2 - Sets the speed of rotation of the spinning cursor when 4D executes formulas. If the value is 0, the spinning cursor does not appear. A positive or negative value sets the speed of rotation to modulo 32. The average value to pass is 16. The default value is 0.
- 3 - Contains the ASCII code for the shortcut generating a new subrecord or record in the active included area of a data entry. The default value is 47 ([/] Key).
- 4 - Contains the value of the modifier key associated with option 3. The default value is 256 (**Command** Key). Possible values include: 256 for the **Command (Ctrl)** Key, 512 for **Shift** Key, 1024 for the **Caps Lock** Key, 2048 for the **Option (Alt)** Key.
- 5 - Contains the ASCII code for the shortcut generating the validation of a data entry. The default value is 3 (**Enter** Key).
- 6 - Contains the value indicating the modifier key associated with the character indicated by option 5. The default value is 0 (no modifier key).
- 7 - Contains the ASCII code for the shortcut generating the cancellation of a data entry. The default value is 46 ([.] Key).
- 8 - Contains the value indicating the modifier key associated with option 7. The default value is 256 (**Command** Key).
- 9 - Contains the ASCII code for the validation shortcut. The default value is 3 (**Enter** Key).
- 10 - Contains the value indicating the modifier key associated with option 9. The default value is 0 (no modifier key).
- 11 - Contains the ASCII code for the shortcut generating the cancellation of a dialog box. The default value is 46 ([.] Key).
- 12 - Contains the value indicating the modifier key associated with option 11. The default value is 256 (**Command** Key).

- 13 - Indicates the default positioning of the main 4D window. If the value is 0, the window is set to the size of the screen with its title bar visible under the menu bar. If the value is 1, the window is set to the size of the screen with its title bar hidden under the menu bar. If it is 2, the window is sized with the coordinates specified by options 14, 15, 16 and 17. If it is 3, the window is sized with the coordinates specified by options 14,15,16, and 17, but is centered in the middle of the screen. Finally, if the value is 4, the window appears with the size and location it had in the previous working session. The default value is 4.

The next four options have a default value equal to -1 and are only used if option 13 contains 2 or 3:

- 14 - Indicates the top coordinate of the 4D window.
- 15 - Indicates the left coordinate of the 4D window.
- 16 - Indicates the bottom coordinate of the 4D window.
- 17 - Indicates the right coordinate of the 4D window.
  
- 18 - Indicates if the main 4D window is resized when you switch to the Runtime environment. If the value is 0, the window is not resized. If it is 1, the window is resized according to option 13. The default value is 0.
  
- 19 - Indicates if the main 4D window is resized after the execution of a 4D method called from a custom menu. If the value is 0, the window is not resized. If it is 1, the window is resized according to option 13. The default value is 0.
  
- 20 - Reserved. Do not modify this option.
  
- 21 - Reserved. Do not modify this option.
  
- 22 - Reserved. Do not modify this option.
  
- 23 - Reserved. Do not modify this option.
  
- 24 - Reserved. Do not modify this option.
  
- 25 - Indicates if the During phase of a form method is called when you cancel a data entry. If the value is 0, the During phase is not triggered, otherwise it is triggered. The default value is 0.
  
- 26 - Indicates if the automatic switch of the read only/read write status of the tables is performed by operations such as display or printing of a selection. If the value is 0, the automatic switch is performed. If it is 1, the automatic switch is not performed. The default value is 0.

Options beyond 26 are reserved. Do not modify them.

**Example**

If you want the cursor to be a spinning wheel when it executes your commands, call:

```
AP SET PARAM(2;16)
```

**See Also**

AP GET PARAM.



# 5

---

## 4D\_Pack : External Clock



The external area \_AP External clock draws an analog clock using the PICT resource 17890 of the 4D\_Pack plug-in. This resource draws the dial of the clock.



You can draw your own clock and then place a 17890 PICT resource in the structure file. This PICT resource can be created on a Macintosh using ResEdit™.

The picture can be of any size, as the size of the clock hands will be adjusted accordingly. In the same manner, the dial shape does not need to be contained in a square, but can be contained in a rectangle.

By default, the clock displays the System time; however, the AP SET CLOCK command allows you to set a delay between the time displayed by the clock and the current time. Thus, you could display different clocks corresponding to different time zones.

**See Also**

AP SET CLOCK.

AP SET CLOCK (clock; hour; minute)

<b>Parameter</b>	<b>Type</b>	<b>Description</b>
clock	Longint	→ _AP External clock external area
hour	Integer	→ Hour delay
minute	Integer	→ Minute delay

### **Description**

The AP SET CLOCK command enables you to set the delay between the time of the clock displayed by the \_AP External clock external area and the current time.

### **See Also**

\_AP External clock.



# 6

---

## 4D\_Pack : Printing



AP Print settings to BLOB (printSettings) → Longint

Parameter	Type		Description
printSettings	BLOB	→	BLOB to store the print settings
Function result	Longint	←	Error code: 1 = Operation successful 0 = No current printer

### Description

The AP Print settings to BLOB command stores the current print settings of 4D in the printSettings BLOB.

The printSettings parameter stores all the settings used for printing:

- Layout parameters: paper, orientation, scale,
- Print parameters as such: number of copies, paper source, etc.

The BLOB generated must not be modified by programming; it can only be used by the AP BLOB to print settings command.

The command returns 1 if the BLOB has been correctly generated and 0 if no current printer is selected.

### See Also

AP BLOB to print settings.

AP BLOB to print settings (printSettings{; paramType}) → Longint

Parameter	Type		Description
printSettings	BLOB	→	BLOB containing the print settings
paramType	Longint	→	0 = Layout and print, 1 = Print
Function result	Longint	←	1 = Operation successful, 0 = No current printer, -1 = Incorrect parameters

### Description

The AP BLOB to print settings command replaces the current print settings of 4D by the parameters stored in the printSettings BLOB. This BLOB must have been generated by the AP Print settings to BLOB command.

The printSettings parameter contains two types of parameters:

- Layout parameters: paper, orientation, scale,
- Print parameters as such: number of copies, paper source, etc.

You can choose to load only the print parameters.

If you pass 0 in paramType or omit this parameter, all the parameters are used. If you pass 1, only the print parameters are used; the layout parameters remain unchanged.

The new print parameters are applied to the entire database and for the whole session, as long as no command such as PAGE SETUP, SET PRINT OPTION or PRINT SELECTION without the > parameter modifies them.

The parameters set are used more particularly by the PRINT SELECTION, PRINT LABEL, PRINT RECORD, Print form and QR REPORT commands, as well as by the menu commands of 4D, including those of the Design environment.

The PRINT SELECTION, PRINT LABEL and PRINT RECORD commands must be called with the > parameter (if applicable) in order for the settings defined by AP BLOB to print settings to be kept.

**Note:** Print settings are not formatted in the same way under Windows and Mac OS. As a result, the compatibility of the printSettings BLOB between the two platforms is not guaranteed.

The command returns 1 if the BLOB has been correctly loaded, 0 if no current printer is selected and -1 if the BLOB is incorrect.

**See Also**

AP Print settings to BLOB.



# 7

---

## 4D\_Pack : ANSI Streams





**Compatibility Note:** This command is now obsolete; it is kept only for compatibility reasons and will be removed in future versions of the plug-in. It is strongly recommended to use the 4D commands of the "Communications" theme.

AP fopen (file | port; mode) → Long integer

Parameter	Type	Description
file   port	String →	Name of the file or port to be opened
mode	Integer →	Opening mode 0 = write only 1 = read only
Function result	Long integer ←	Value to be used with AP FPRINT, or 0 if the file or port cannot be open

### Description

The AP fopen command calls the "fopen" ANSI C command, which enables the opening of serial or parallel ports (on Windows) or the direct creation of files.

The value returned by this function is then used with the AP FPRINT command to write through the port, or to write the file. The opened port or file can then be closed using the command AP FCLOSE.

**Note:** On Mac OS, this function works only with the direct creation of files.

### Examples

1. Writing "hello world" on the parallel port:

```
port:=AP fopen ( "LPT1";0 )
AP FPRINT ( port; "hello world" )
AP FCLOSE ( port )
```

2. Writing "hello world" on the serial port:

```
port:=AP fopen ( "COM1";0 )  
AP FPRINT ( port; "hello world" )  
AP FCLOSE ( port )
```

3. Writing "hello world" in a file:

```
port:=AP fopen ( "C:\myfile.txt";0 )  
AP FPRINT ( port; "hello world" )  
AP FCLOSE ( port )
```

4. Reading a file:

```
port:=AP fopen ( "C:\myfile.txt";1 )  
MyVar:=AP fread ( port )  
AP FCLOSE ( port )
```

### **See Also**

AP FPRINT.

---

**Compatibility Note:** This command is now obsolete; it is kept only for compatibility reasons and will be removed in future versions of the plug-in. It is strongly recommended to use the 4D commands of the "Communications" theme.

AP FPRINT (fopenValue; data)

Parameter	Type	Description
fopenValue	Longint	→ Port or file reference value returned by AP fopen
data	Text	→ Data to be sent to the port or file

### Description

The AP FPRINT command sends the specified data text to the port or file opened by the AP fopen command whose reference is passed in fopenValue.

data is sent as it is without any conversion. Since 4D uses the Macintosh ASCII table internally, you must perform the required conversion on Windows. To do so, use the Mac to Win 4D command.

### See Also

AP FCLOSE, AP fopen, Mac to Win.

**Compatibility Note:** This command is now obsolete; it is kept only for compatibility reasons and will be removed in future versions of the plug-in. It is strongly recommended to use the 4D commands of the "Communications" theme.

AP FCLOSE (fopenValue)

<b>Parameter</b>	<b>Type</b>	<b>Description</b>
fopenValue	Longint	→ Port or file reference value returned by AP fopen

### **Description**

The AP FCLOSE command closes the port or file previously opened using the AP fopen command.

### **See Also**

AP fopen, AP FPRINT.

**Compatibility Note:** This command is now obsolete; it is kept only for compatibility reasons and will be removed in future versions of the plug-in. It is strongly recommended to use the 4D commands of the "Communications" theme.

AP fread (fopenValue) → Text

Parameter	Type		Description
fopenValue	Longint	→	Port or file reference value returned by AP fopen
Function result	Text	←	Data from the port or file

### Description

AP fread reads the data from the port or file opened with AP fopen whose reference is passed in fopenValue and returns it as text.

Data is returned without any conversion. Since 4D uses the Macintosh ASCII table internally, you must perform the required conversion on Windows. To do so, use the Win to Mac 4D command.

### See Also

AP fopen, Win to Mac.



# 8

---

## 4D\_Pack : Picture Files





**Compatibility Note:** This command is now obsolete; it is kept only for compatibility reasons and will be removed in future versions of the plug-in. It is strongly recommended to use the 4D commands of the "Pictures" theme.

AP Save BMP 8 bits (image; file) → Longint

Parameter	Type		Description
image	Picture	→	Name of a 4D picture variable
file	Text	→	Name of the file in which the BMP will be saved
Function result	Longint	←	0, if the operation was successful; otherwise a system error code

### Description

AP Save BMP 8 bits saves any 4D picture variable in a regular 8-bit BMP picture (256 colors and no compression). If an identically named file is located where you want to save the picture, it is replaced without warning. This command works on both Macintosh and Windows platforms.

If the operation is successfully completed, the function returns 0; otherwise, it returns a negative error code.



# Command Index

\_AP External clock.....47

## A

AP AVAILABLE MEMORY..... 38

## B

AP BLOB to print settings.....52

## C

AP CLOSE HELP.....20

AP Create method..... 29

## D

AP Does method exist.....31

## F

AP FCLOSE.....60

AP fopen.....57

AP FPRINT.....59

AP fread.....61

AP FULL SCREEN..... 12

## G

AP Get field infos.....	26
AP Get file MD5 digest.....	35
AP GET PARAM.....	39
AP Get picture type.....	37
AP Get table info.....	25
AP Get templates.....	28
AP Get tips state.....	33

## H

AP HELP INDEX.....	18
AP HELP ON HELP.....	19
AP HELP ON KEY.....	17

## M

AP Modify method.....	32
-----------------------	----

## N

AP NORMAL SCREEN.....	13
-----------------------	----

## P

AP Print settings to BLOB.....	51
--------------------------------	----

## R

AP Rect dragger.....	11
----------------------	----

## S

AP Save BMP 8 bits.....	65
AP SET CLOCK.....	48
AP SET PARAM.....	40
AP SET TIPS STATE.....	34
AP ShellExecute.....	23

## T

AP Timestamp to GMT.....	36
--------------------------	----

